

**COMPUTER PROGRAMS FOR
COUNTERCURRENT DISTRIBUTION**

A recent publication* considered nonideal phenomena encountered in counter-current distribution. We found it necessary to simulate these processes on the computer. Two programs (I and II) were written in FORTRAN IV* for an IBM 1130** equipped with card reader-punch, disk, 8 K core.

PROGRAM I

Program I is less fundamental than Program II, but it is a good approximation under the following conditions: The solutes effectively have no volume; there is no solute-solute interaction; the partition coefficients are constant or vary as a function of total concentration; solvent volumes are constant. These assumptions are reasonable in dilute solutions.

Two simulations may be done simultaneously. The partition coefficient and concentrations for the first solute are designated by variables in the program beginning with the letter C. The variables associated with the second solute begin with the letter X. An apparatus of 1 to 200 tubes may be simulated. The number of tubes in the apparatus and the volume information for both systems are the same.

For each system, one or more inputs of solute may be desired. An elution profile results from a single input of solute. If enough increments of solute are added to tube zero, the maximum equilibrium concentration will be reached in the last tube and a frontal output profile will result. A number of inputs between two and the number required to just reach the maximum equilibrium concentration will give an output profile somewhere between an elution and a frontal. This region may be interesting in certain systems, and it is particularly suited to computer simulation since this intermediate region is difficult to describe mathematically.

Solute may be introduced to the instrument by two different procedures. If data switch '14' is on, the solute is placed in one or more tubes before the simulation is started. This batch loading reduces the number of effective tubes in the instru-

*Rothbart, H. L., Barford, R. A., Martin, V. G., Bertsch, R. J., and Eddy, C. R. Nonideal Phenomena in Countercurrent Distribution. Separation Science [In press.]

**Trade names are used in this publication solely for the purpose of providing specific information. Mention of a trade name does not constitute a guarantee or warranty of the product by the U. S. Department of Agriculture or an endorsement by the Department over other products not mentioned.

ment. If data switch '14' is off, the solute is added to the zeroth tube in one or more increments with a transfer occurring between each addition. This sequential addition of solute gives a better separation than batch loading for the same number of tubes.

The partition coefficients may be stated at the beginning of the simulation and assumed constant, or a subroutine JVARY may be written to vary the partition coefficients as a function of solute concentration. The example of JVARY in this publication shows the partition coefficients (CK, XK) varying as a linear function of total amount of solute (T2, XT2) in a tube. CK increases with solute concentration and XK decreases. There is nothing fundamental about a linear variation and other functions will be more suitable in some systems.

In a countercurrent distribution, the lower phase volume is not always equal to the cut-off volume of a tube. If the lower phase is above the cut-off point, some "stationary" phase will be transported along with the mobile phase. Lower phase can also be held at the air-liquid interface and thus be transported. If the lower phase is below the cut-off point, some of the mobile phase will be retained. Mobile phase can also be retained on the walls of the tubes*. For these reasons four volume terms are needed to describe a simulation: total upper volume, total lower volume, volume upper retained, volume lower retained. These four terms are input parameters.

During a simulation, the amount of overlap between two solute profiles may be calculated. This option is selected by turning on data switch '5'. The tubes in the apparatus are separated into two groups at the point where the two curves cross. The contents of all tubes to the left of the crossing point are collected as the first fraction. The contents of all tubes to the right are collected as the second fraction. Tubes with solute amounts less than 0.0001 are ignored. The tube number and height of each peak are found. The weights of both components in each fraction are calculated. The percent impurity of a fraction is defined as: weight of lesser solute/total weight of both solutes. The total impurity is the sum of the two percent impurities and is also in percent units. The above computations are printed for each transfer. However, if data switch '13' is on, the impurity calculations are under the same control as the main printout.

Program I is in two sections. The first section, called SEP62, reads the input parameters and performs the simulation, writing the output profiles on the disk. Since this calculation can be quite time-consuming, the program can be interrupted and restarted at a later time. Data switch '3' will interrupt the calculation in this manner, causing the restart information to be saved on the disk. If 200 tubes are utilized in a simulation, approximately 350 transfers an hour will be performed. The first 10 transfers are automatically printed and then transfers 50, 100, 150, and so forth are printed. The operator can override the automatic print control if desired. While data switch '11' is on, current transfers are printed. Data switch '12' on will suppress all printout. Switch '12' is especially useful when only the output is of interest.

The output information from SEP62 consists of transfer number, tube number, and for each of solute systems 1 and 2: total amount of solute, concentration in the upper phase, concentration in the lower phase. Tube numbering begins at zero; for example, in a 200-tube simulation the tubes will be numbered from 0 to 199. Transfer number also begins at zero, that is, operation 1 is labeled transfer number 0.

The second section of Program I, called VEP62, reads the output profile from the disk and writes it on the printer. For simulations involving a large number of transfers only every 2d, 5th, and 10th transfer may be of interest. The program reads one card with the increment between transfer number. If the card is blank, every transfer will be printed. No information will be printed unless the total solute eluted is greater than 0.0001. The percent impurity calculation is like that in SEP62. This calculation is automatically printed unless data switch '5' is on.

The output profile may be punched on cards for plotting or other uses. To punch the profile from the first partition coefficient (CK-system), turn on data switch '1'; from the second partition coefficient (XK), turn on data switch '2'. Both switches may be on simultaneously. The cards of the XK output profile will come out in the alternate stacker. Solute amounts less than 0.0001 will not be punched.

Card Input Required for SEP62

				<u>Format</u>
Card 1				
Number of transfers	Number of tubes			2I5
Card 2				
Total upper volume	Total lower volume	Volume upper retained	Volume lower retained	4F10.5
Card 3				
K _D first solute	Grams per increment first solute	Number of increments first solute		2F10.5, I5
Card 4				
K _D second solute	Grams per increment second solute	Number of increments second solute		2F10.5, I5

Sample Input for SEP62

500	200		
20.0	39.2	1.4	38.6
5.2	1.0	50	
7.6	2.0	35	

Increment
for printout

Correlation of Systems of Symbols

<u>Item</u>	<u>Program</u>	<u>Reference*</u>
Partition coefficient	CK, XK	K _D
Upper phase volume	VUPPR	V _U
Lower phase volume	VLOW	V _L
Number of tubes in apparatus	NTUBE	p
Number of increments of solute	NOFF1, NOFF2	n
Upper volume transferred	VUPUL	V _{UT}
Lower volume transferred	VLPUL	V _{LT}

Summary of Switches Used in SEP62

- '0' continues with previously interrupted calculation
- '3' interrupts calculation
- '5' calculates and prints percent impurities
- '11' prints every transfer
- '12' suppresses all printout
- '13' places impurities printout under control of automatic printout and subject to control by switches '11' and '12'
- '14' causes batch rather than sequential loading of solute

Summary of Switches Used in VEP62

- '1' punches output profile of the CK system
- '2' punches output profile of the XK system
- '5' suppresses percent impurity calculation

*See reference in footnote p. 1.

PROGRAM I

MAINLINE PROGRAM SEP62

```

*EXTENDED PRECISION
*ONE WORD INTEGERS
*I0CS(CARD,1132PRINTER,DISK,TYPEWRITER)
** VERSION OF AUGUST 5,1969 V.G. MARTIN
C    BUFFR =      ARRAY CONTAINING CONCENTRATIONS.
C    CK     =      PARTITION COEFFICIENT (XK)
C    CLO1,2=      CONCENTRATION IN LOWER PHASE (XCLO1,2)
C    CUP1,2=      CONCENTRATION IN UPPER PHASE (XCUP1,2)
C    FILE2 =      ARRAY FOR OUTPUT PROFILE.
C    JR     =      TUBE NUMBER PLUS ONE.
C    KREC   =      RECORD NUMBER IN FILE 2 - APPEARS IN DEFINE FILE
C                  STATEMENT.
C    N      =      TRANSFER NUMBER PLUS ONE
C    N1    =      LOWER BOUND OF N IN TRANSFER NUMBER LOOP.
C    N25   =      ARGUMENT 3 IN SUBROUTINE MOD. IT EQUALS THE INTEGER
C                  PART OF ARGUMENT 1 DIVIDED BY ARGUMENT 2.
C    NEXIT  =      INDICATOR FOR SWITCH 3.
C    =          1 EXIT AFTER PRINTING OUT NEXT COMPLETE JR-LOOP.
C    =          2 CONTINUE WITH CALCULATION.
C    NLIM   =      UPPER BOUND OF N IN TRANSFER NUMBER LOOP.
C    NOFF1  =      NUMBER OF SOLUTE INPUTS (NOFF2)
C    NOUT   =      SUBSCRIPT WITHIN RECORD OF FILE 2 = NEXT AVAILABLE
C                  POSITION.
C    NPRNT  =      PRINT CONTROL
C    =          1 PRINT
C    =          2 DON'T PRINT
C    NREC   =      RECORD NUMBER IN FILE 1.
C    NTUBE  =      NUMBER OF TUBES IN APPARATUS.
C    T2     =      TOTAL SOLUTE IN A TUBE (XT2)
C    TINT1  =      GRAMS OF SOLUTE FED ON ONE INPUT (TINT2)
C    VLOW   =      LOWER VOLUME TOTAL.
C    VLPUL  =      VOLUME LOWER PULSED.
C    VUPPR  =      UPPER VOLUME TOTAL.
C    VUPUL  =      VOLUME UPPER PULSED.
C    VLRET  =      VOLUME LOWER RETAINED.
C    VURET  =      VOLUME UPPER RETAINED.

      INTEGER PEAKC,PEAKX
      DIMENSION FILE2(104), BUFFR(100)
      DEFINE FILE 1(9,320,U,NREC)
      DEFINE FILE 2(300,320,U,KREC)
5       FORMAT('1 TRANS TUBE' 13X'TOTAL'5X'CONC. UP'3X'CONC. LOW'11X'TOTA
-L'5X'CONC. UP'3X'CONC. LOW')
10      FORMAT(' TURN ON SWITCH 0 TO CONTINUE OLD RUN'/
-' TURN ON SWITCH 5 FOR PERCENT IMPURITIES CALCULATION'/
-' TURN ON SWITCH 14 FOR BATCH LOADING')
15      FORMAT (2I5)
20      FORMAT (4F10.5)
25      FORMAT (2F10.5, 15)
30      FORMAT(17,16,7X,3F12.6,5X,3F12.6)
35      FORMAT(//16,2X,'TUBES')
40      FORMAT(16,2X'TRANSFERS')

```

PROGRAM I

SEP62 (CONT.)

```

45   FORMAT ('UPPER VOLUME'F12.2)
50   FORMAT(' LOWER VOLUME'F12.2)
55   FORMAT(' UPPER TRANSFERED' F8.2)
60   FORMAT(' LOWER TRANSFERED' F8.2)
65   FORMAT (///'OPARTITION COEFF.'5X'FEED WT.'3X'NUMBER OF TUBES INITI
-ALLY'/37X'FILLED WITH SOLUTE')
70   FORMAT(///'O PARTITION COEFF.'5X'FEED WT.'3X'NUMBER OF INPUTS')
75   FORMAT(F12.3,F18.3,6X,17)
80   FORMAT('OTRANSFER' 15)
85   FORMAT(' PEAK1 LOCATION ='14' HEIGHT ='F7.4,7X'PEAK2 LOCATION =
A'14' HEIGHT ='F7.4)
90   FORMAT(' FRACTION 1'5X'TUBES'14'-'13,17X'FRACTION 2'5X'TUBES'14'-
A13)
95   FORMAT(' WEIGHT C-COMPONENT ='F9.4,16X'WEIGHT C-COMPONENT ='F9.4)
100  FORMAT(' WEIGHT X-COMPONENT ='F9.4,16X'WEIGHT X-COMPONENT ='F9.4)
105  FORMAT(' PERCENT IMPURITY ='F9.4,16X'PERCENT IMPURITY ='F9.4)
110  FORMAT(21X'TOTAL IMPURITY ='F9.4)
110
MAXNO = 0
NEXIT = 2
WRITE(1,10)
PAUSE 7777
CALL DATSW(5,MPURE)
CALL DATSW (14,JSW14)
C      SWITCH 0 -- OFF- START NEW RUN.  ON-- CONTINUE OLD RUN.
CALL DATSW(0,MR)
GO TO (115,120),MR
115  READ(1'9)N,NLIM,NTUBE,KREC,NOUT,VUPUL,VLPUL,VURET,VLRET,CK,XK,TINT
    11,TINT2,NOFF1,NOFF2
    NO1 = NOFF1
    NO2 = NOFF2
    READ(2'KREC)FILE2
    KREC = KREC - 1
    N1 = N + 1
    VUPPR = VUPUL + VURET
    VLOW = VLPUL + VLRET
    NTUBE = NTUBE - 1
    GO TO 195
120  READ(2,15)NLIM,NTUBE
    READ(2,20)VUPPR,VLOW,VURET,VLRET
    VUPUL = VUPPR-VURET
    VLPUL = VLOW - VLRET
    READ(2,25)CK,TINT1,NOFF1
    READ(2,25)XK,TINT2,NOFF2
    NO1 = NOFF1
    NO2 = NOFF2
    DO 125 KR=1,100
    125  BUFFR(KR) = 0.0
    DO 130 KR=1,8
    130  WRITE(1'KR)BUFFR
    GO TO (135,195),JSW14
    135  CUP1 = TINT1 / (VUPPR + VLOW/CK)

```

PROGRAM I

SEP62 (CONT.)

```
CLO1 = CUP1 / CK
XCUP1= TINT2/(VUPPR + VLOW/XK)
XCL01 = XCUP1 / XK
IF(NOFF1-NOFF2) 140,145,145
140 MAXNO = NOFF2
MINNO = NOFF1
GO TO 150
145 MAXNO = NOFF1
MINNO = NOFF2
150 DO 190 I=1,MAXNO
IF(I-MINNO) 170,170,155
155 IF(NOFF1-NOFF2) 160,165,165
160 CUP1 = 0.0
CLO1 = 0.0
GO TO 170
165 XCUP1 = 0.0
XCL01 = 0.0
170 J=1
CALL MOD (J,25,N25)
IF(J) 175,175,180
175 J = 25
GO TO 185
180 N25 = N25 + 1
185 READ(1'N25)BUFFR
BUFFR (J) = CUP1
BUFFR (J+ 25) = CLO1
BUFFR (J+ 50) = XCUP1
BUFFR (J+75) = XCL01
190 WRITE(1'N25)BUFFR
NOFF1 = 0
NOFF2 = 0
195 WRITE(3,35)NTUBE
WRITE(3,40)NLIM
WRITE(3,45)VUPPR
WRITE(3,50)VLOW
WRITE(3,55)VUPUL
WRITE(3,60)VLPUL
GO TO (200,205),JSW14
200 WRITE(3,65)
GO TO 210
205 WRITE(3,70)
210 WRITE(3,75)CK,TINT1,N01
WRITE(3,75)XK,TINT2,N02
GO TO (220,215),MR
C      INITIALIZATION STEPS FOR NEW RUN.
215 KREC = 1
NOUT = 1
N1 = 1
220 NTUBE = NTUBE + 1
ISIGN = 1
IF(CK-XK) 225,230,230
```

PROGRAM I

SEP62 (CONT.)

PROGRAM I

SEP62 (CONT.)

```

C           SOLUTE 2 AMT UPR, AMT LOW ELUTED
FILE2(NOUT) = VUPUL * BUFR(KR)
FILE2(NOUT+26) = VLPUL * BUFR(KR+25)
FILE2(NOUT+52) = VUPUL * BUFR(KR+50)
FILE2(NOUT+78) = VLPUL * BUFR(KR+75)
KR = NOUT
CALL MOD (KR,26,N25)
IF(KR) 325,325,330
325  WRITE(2'KREC)FILE2
NOUT = 0
330  NOUT = NOUT + 1
GO TO 470
335  NREC = N25 + 1
READ(1'NREC)BUFR
C           RECYCLING THE JR LOOP BEGINS HERE.
340  KR = JR
CALL MOD(KR,25,N25)
IF(KR-1) 345,350,365
345  NREC = N25 + 1
WRITE(1'NREC)BUFR
READ(1'N25)BUFR
KR = 25
GO TO 365
350  NREC = N25 + 1
IF(JR-1) 355,355,360
355  CUP1 = TINT1/VUPUL
XCUP1 = TINT2/VUPUL
CLO1 = 0.0
XCLO1 = 0.0
GO TO 370
360  WRITE(1'NREC)BUFR
READ(1'N25)BUFR
CUP1 = BUFR(25)
CLO1 = BUFR(50)
XCUP1 = BUFR(75)
XCLO1 = BUFR(100)
READ(1'NREC)BUFR
GO TO 370
365  CUP1 = BUFR(KR-1)
CLO1 = BUFR(KR+24)
XCUP1 = BUFR(KR+49)
XCLO1 = BUFR(KR+74)
370  CUP2 = BUFR(KR)
CLO2 = BUFR(KR+25)
XCUP2 = BUFR(KR+50)
XCLO2 = BUFR(KR+75)
T2 = CUP1*VUPUL+CLO1*VLPUL+CUP2*VURET+CLO2*VLRET
XT2 = XCUP1*VUPUL+XCLO1*VLPUL+XCUP2*VURET+XCLO2*VLRET
GO TO (375,450),MPURE
375  IF(T2-TMAX) 385,380,380
380  TMAX = T2

```

SEP62 (CONT.)

```
PEAKC = JR-1
385 IF(XT2-XTMAX) 395,390,390
390 XTMAX = XT2
391 PEAKX=JR-1
395 DEL = (T2-XT2)*ISIGN
SUMC = SUMC + T2
SUMX = SUMX + XT2
GO TO (435,400),KDEL
400 IF(DEL) 405,420,420
405 KDEL = 1
IF(N4) 410,410,415
410 N2 = JR-1
GO TO 450
415 N3 = JR
N2 = JR-1
WTC = SUMC - T2
WTX = SUMX - XT2
SUMC = T2
SUMX = XT2
GO TO 450
420 IF(T2+XT2-.0001) 450,450,425
425 IF(N4) 430,430,450
430 N4 = JR-1
GO TO 450
435 IF(T2+XT2-.0001) 440,440,450
440 IF(N1) 445,445,450
445 N1 = JR
450 CALL JVARY(T2,XT2,CK,XK)
CLO2 = T2 / (VLOW+CK*VUPPR)
CUP2 = CK * CLO2
XCLO2 = XT2 / (VLOW+XK*VUPPR)
XCUP2 = XK * XCLO2
BUFFR(KR) = CUP2
BUFFR(KR+25) = CLO2
BUFFR(KR+50) = XCUP2
BUFFR(KR+75) = XCLO2
CALL DATSW (12,MR)
GO TO (455,460),MR
455 NPRNT = 2
460 GO TO (465,470),NPRNT
465 M = N - 1
KR = JR - 1
WRITE(3,30)M,KR,T2,CUP2,CLO2,XT2,XCUP2,XCLO2
470 JR = JR - 1
IF(JR) 475,475,340
475 WRITE(1'1)BUFFR
GO TO (480,555),MPURE
480 CALL DATSW(13,JSW13)
GO TO (485,490),JSW13
485 GO TO (490,555),NPRNT
```

SEP62 (CONT.)

```
490    M = N - 1
      WRITE(3,80)M
      IF(N4) 495,495,530
495    IF(ISIGN) 500,520,520
500    IF(TMAX-.0001) 505,505,510
505    MPURE = 2
      GO TO 555
510    WRITE(3,85)PEAKC,TMAX
      SUM1 = SUMX
515    WRITE(3,90)N1,N2
      WRITE(3,95)SUMC
      WRITE(3,100)SUMX
      FIMP = SUM1*100./(SUMC+SUMX)
      WRITE(3,105)FIMP
      GO TO 550
520    IF(XTMAX-.0001) 505,505,525
525    WRITE(3,85)PEAKX,XTMAX
      SUM1 = SUMC
      GO TO 515
530    IF(ISIGN) 535,545,545
535    WRITE(3,85)PEAKC,TMAX,PEAKX,XTMAX
      SUM1 = SUMX
      SUM2 = WTC
540    WRITE(3,90)N1,N2,N3,N4
      WRITE(3,95)SUMC,WTC
      WRITE(3,100)SUMX,WTX
      FIMP = SUM1 * 100./(SUMX+SUMC)
      GIMP = SUM2*100./(WTX+WTC)
      WRITE(3,105)FIMP,GIMP
      FIMP = FIMP+GIMP
      GO TO 550
545    WRITE(3,85)PEAKX,XTMAX,PEAKC,TMAX
      SUM1 = SUMC
      SUM2 = WTX
      GO TO 540
550    WRITE(3,110)FIMP
555    GO TO (570,560),NEXIT
C     SWITCH 3 - INTERRUPT CALCULATION.
560    CALL DATSW (3,KR)
      GO TO (565,575),KR
565    NEXIT = 1
      GO TO 575
570    WRITE(1'9)N,NLIM,NTUBE,KREC,NOUT,VUPUL,VLPUL,VURET,VLRET,CK,XK,TIN
      1T1,TINT2,NOFF1,NOFF2
      WRITE(2'KREC)FILE2
      CALL EXIT
575    CONTINUE
      N = N - 1
      GO TO 570
END
```

PROGRAM I

SUBROUTINE MOD

```
*EXTENDED PRECISION
*ONE WORD INTEGERS
C      V. G. MARTIN
      SUBROUTINE MOD(NUMBR,INCRE,MANY)
C      CALCULATE ARGUMENT 1 MODULO ARGUMENT 2.
      MANY = NUMBR / INCRE
      NUMBR = NUMBR - MANY * INCRE
      RETURN
      END
```

SUBROUTINE JVARY

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      V. G. MARTIN
      SUBROUTINE JVARY(T2,XT2,CK,XK)
C-----VARY THE PARTITION COEFFICIENTS AS A FUNCTION OF TOTAL
C      CONCENTRATION OF SOLUTE.
      CK = .1077 + .0007* T2
      XK = .1077 - .0007 * XT2
      RETURN
      END
```

MAINLINE PROGRAM VEP62

```
*EXTENDED PRECISION
*ONE WORD INTEGERS
*I0CS(CARD,1132PRINTER,DISK)
**      VERSION OF MAY 26, 1969      V.G. MARTIN
      INTEGER PEAKC,PEAKX
      DIMENSION FILE2(104)
      DEFINE FILE 1( 9,320,U,NREC)
      DEFINE FILE 2(300,320,U,KREC)
5      FORMAT (16,2E13.5,2X,2E13.5)
10     FORMAT('OTRANSFER'5X'TOTAL'7X'UPPER'10X'TOTAL'7X'UPPER'//)
15     FORMAT (13)
20     FORMAT (2X,2F13.8)
25     FORMAT('OPEAK1 LOCATION ='14' HEIGHT ='F7.4,7X'PEAK2 LOCATION =
-'14' HEIGHT ='F7.4)
30     FORMAT(' FRACTION 1'5X'TUBES'14'-'13,17X'FRACTION 2'5X'TUBES'14'-
-13)
35     FORMAT(' WEIGHT C-COMPONENT ='F9.4,16X'WEIGHT C-COMPONENT ='F9.4)
40     FORMAT(' WEIGHT X-COMPONENT ='F9.4,16X'WEIGHT X-COMPONENT ='F9.4)
45     FORMAT(' PERCENT IMPURITY   ='F9.4,16X'PERCENT IMPURITY   ='F9.4)
```

PROGRAM I

VEP62 (CONT.)

```

50   FORMAT(21X'TOTAL IMPURITY ='F9.4)
      CALL DATSW (1,JSW1)
      CALL DATSW (2,JSW2)
      CALL DATSW(5,MPURE)
      READ(2,15)INCR
      IF(JSW1 + JSW2 - 4) 52, 53, 53
52   READ (2,15)
53   IF(INCR) 55, 55, 60
55   INCR = 1
60   WRITE(3,10)
      READ(1'9)N,NLIM,NTUBE,KREC,NOUT,V,V,V,V,V,CK,XK
      GO TO (80,65),MPURE
65   ISIGN = 1
      KDEL = 2
      IF(CK-XK) 70,75,75
70   ISIGN = -1
75   SUMC = 0.0
      SUMX = 0.0
      TMAX = 0.0
      XTMAX= 0.0
      N1 = 0
      N2 = 0
      N3 = 0
      N4 = 0
80   K = NTUBE - 2
      KREC = 1
      DO 320 I=NTUBE,N,26
      READ(2'KREC)FILE2
      DO 315 J=1,26
      K = K + 1
      K1 = K / INCR
      K1 = K1 * INCR - K
85   IF(K-N) 90,325,325
90   TUP = FILE2(J)
      TLO = FILE2(J+26)
      T2 = TUP+TLO
      XTUP = FILE2(J+52)
      XTL0 = FILE2(J+78)
      XT2 = XTUP + XTL0
      TEST = T2 + XT2
      IF(TEST-.0001) 95,95,125
95   IF(N1) 315,315,100
100  IF(N2) 105,105,110
105  N2 = K - 1
      GO TO 315
110  IF(N3) 315,315,115
115  IF(N4) 120,120,315
120  N4 = K - 1
125  IF(K1) 160,130,160
130  WRITE(3,5)K,T2,TUP,XT2,XTUP

```

VEP62 (CONT.)

```
          GO TO (135,145),JSW1
135    XK = K
          IF(T2-1.0E-04) 145,145,140
140    WRITE(2,20)XK,T2
145    GO TO (150,160),JSW2
150    XK = K
          IF(XT2-1.0E-04) 160,160,155
155    WRITE(2,20)XK,XT2
          CALL STACK
160    GO TO (310,165),MPURE
165    IF(T2-TMAX) 175,170,170
170    TMAX = T2
          PEAKC = K
175    IF(XT2-XTMAX) 185,180,180
180    XTMAX = XT2
          PEAKX = K
185    DEL = (T2-XT2)*ISIGN
          SUMC = SUMC + T2
          SUMX = SUMX +XT2
          GO TO (190,230),KDEL
190    IF(ISIGN) 195,225,225
195    IF(T2-.0001) 200,200,215
200    IF(N3) 310,310,205
205    IF(N4) 210,210,310
210    N4 = K - 1
          GO TO 310
215    IF(N3) 220,220,310
220    N3 = K
          GO TO 310
225    IF(XT2-.0001) 200,200,215
230    IF(DEL) 275,235,235
235    IF(ISIGN) 270,240,240
240    IF(T2-.0001) 245,245,260
245    IF(N1) 310,310,250
250    IF(N2) 255,255,310
255    N2 = K-1
          GO TO 310
260    IF(N1) 265,265,310
265    N1 = K
          GO TO 310
270    IF(XT2-.0001) 245,245,260
275    KDEL = 1
          IF(N2) 280,280,285
280    N2 = K - 1
          N3 = K
          GO TO 300
285    IF(ISIGN) 290,290,305
290    IF(T2-.0001) 300,300,295
295    N3 = K
300    WTC = SUMC - T2
```

PROGRAM I

VEP62 (CONT.)

```
WTX = SUMX - XT2
SUMC = T2
SUMX = XT2
GO TO 310
305 IF(XT2-.0001) 300,300,295
310 CONTINUE
315 CONTINUE
320 CONTINUE
325 GO TO (350,330),MPURE
330 IF(ISIGN) 340,340,335
335 WRITE(3,25)PEAKC,TMAX,PEAKX,XTMAX
SUM1 = WTX
SUM2 = SUMC
GO TO 345
340 WRITE(3,25)PEAKX,XTMAX,PEAKC,TMAX
SUM1 = WTC
SUM2 = SUMX
345 WRITE(3,30)N1,N2,N3,N4
WRITE(3,35)WTC,SUMC
WRITE(3,40) WTX,SUMX
FIMP = SUM1 * 100./ (WTC+WTX)
GIMP = SUM2 * 100./ (SUMC+SUMX)
WRITE(3,45)FIMP,GIMP
FIMP = FIMP + GIMP
WRITE(3,50)FIMP
350 CALL EXIT
END
```

PROGRAM II

Program II performs a countercurrent simulation of one solute. Information from the ternary diagram of this solute and the two solvents used in the distribution is incorporated in the computer simulation. Many of the limitations of Program I are thus overcome. Program II allows for solute volume, solute-solvent interaction, mutual solubilities of the solvents, complex variation in the partition coefficient, volume reorganization, and isopycnic phenomena*. All data particular to a certain ternary system are isolated in subroutines so the mainline programs need not be recompiled for each separate system. Program II is general for most Type I and Type II liquid-liquid systems, ** provided the equilibria are expressed in the following manner:

1. Triangular coordinates are used and the units are expressed in weight fraction.
2. The solute is the independent variable.
3. The best solvent for the solute is the dependent variable (Y). This convention makes the tie line slopes $\Delta Y / \Delta X$ positive. Solutropes, therefore, cannot be processed with this program.
4. The other solvent (Z) is not specified but can always be calculated by $Z = 1 - X - Y$, since the weight fractions add up to 1.0.
5. Tie lines are described by their slopes and Y-intercepts in a subroutine SBTIE.
6. The solubility curves are expressed as polynomials in subroutines CURVU, CURVL, KURVU, KURVL. The U and L symbolize the upper and lower curves on the triangular plot, not necessarily upper and lower phase.

The computation is done by four mainline programs: TERN1, TERN2, TERN3 perform the simulation, while PUGET reads and prints the output profile. TERN1, TERN2, and TERN3 need not be distinct segments on a computer with larger core capacity. The main computation is interruptable by switch '6'.

If the program is manually interrupted by switch '6', TERN3 is called and saves the data on the disk. The next time the program is executed, this interrupted run will automatically continue until interrupted again with switch '6', terminated with switch '3', or the desired number of transfers are completed. In the latter two instances, it is not possible to restart an old run and an execution will result in reading a set of data and starting a new computation.

*See reference in footnote p. 1.

**Treybal, R. E. Liquid extraction. 1963. Ed. 2. McGraw Hill, New York.

In segment TERN1, the input cards are interpreted and initial tube conditions are established. Other necessary input parameters are also set. In the case of a restart, cards are not read but the information is taken off the disk.

TERN2 performs the actual simulation. The total composition in each tube is calculated. The composition point is tested to see if it is in the miscible region (block 3). If so, the two tie lines between which the point lies are found and their intersection calculated. A new tie line is drawn between this intersection and the point in question. The intersections of the new tie line and the solubility curves are found (block 4). These intersections are the compositions in weight fractions of the upper and lower phases. The densities of the two phases are calculated and tested for phase inversion (block 5). The weight of each phase is calculated by the lever rule. The volumes are determined and a transfer is made correcting for volumes retained on the tube walls (block 6).

PUGET is a mainline program for printing the output profile. This program also uses data from the ternary diagram. It performs essentially the same set of calculations as TERN2.

Input

	Format
Card 1	
Print control	KPRNT array
	16I5
	transfers from KPRNT (N) → KPRNT (N+1)
	will be automatically printed, N = 1, 3, 5,...15
Card 2	
Mnemonics for the three components	3 (A3, 1X)
Card 3	
Number of transfers, number of tubes, number of inputs	3I5
Card 4	
Densities of: X, Y, Z, most dense solvent, least dense solvent	5F7.4
Card 5	
Upper, lower volumes retained on the wall	2F5.2
Card 6	
Feed solution information	19X, I1, 4F10.4
	NGOTO, XUP1, XUP2, XUP3, VIN

There are three choices for specifying
the feed solution

NGOTO = 1

XUP1 is interpreted as grams of solute per input

XUP2 is volume of pre-equilibrated upper phase to be fed

NGOTO = 2

XUP1 is weight of solute (X)

XUP2 is weight of Y

XUP3 is weight of Z

NGOTO = 3

XUP1 is weight fraction of X in feed solution

XUP2 is weight fraction of Y in feed solution

XUP3 is weight fraction of Z in feed solution

VIN is the volume of upper phase to be fed

Card 7

Initial tube condition information

19X, I1, 4F10.4, 2I5

NGOTO, XLOW1, XLOW2, VLOW, VECUT, NTUBA, NTUBZ

There are two choices for specifying initial tube
conditions

NGOTO = 1

XLOW1 is weight of solute (X) initially in
tubes NTUBA → NTUBZ

VLOW is volume of lower phase in
tubes NTUBA → NTUBZ

VECUT is cut-off volume of tubes
NTUBA → NTUBZ

NGOTO = 2

XLOW1 is weight of solute (X) in tubes
NTUBA → NTUBZ

XLOW is weight of Z in tubes NTUBA → NTUBZ

VLOW is weight of Y in tubes NTUBA → NTUBZ

VECUT is the cut-off volumes of tubes NTUBA → NTUBZ

If option 2 is selected, some tubes can be filled with both phases initially.

Depending on the choice of NTUBA and NTUBZ, many different number 7 cards may be read for a single run. This permits partial batch loading of the instrument, different cut-off volumes for every tube, and other flexibilities. However, small random deviations in the cut-off volumes of tubes such as occur in a real apparatus have been shown to produce little change in a distribution*. The computer will continue reading '7' cards until a value of NTUBZ corresponding to the serial number of the last tube +2 is encountered. A value of 2 for NTUBA corresponds to the first tube in the apparatus. For example, if tubes 0-199 were to be filled NTUBA = 2, NTUBZ = 201.

Card 8

08888

I5

If you wish to do another run immediately, omit the number 8 card and insert new set of data starting with print control.

Sample Input

0 25 50 50 100 100 150 150 200 200 250 250 275 275 300 300

CMX CMY CMZ

180 10 100

.7870 1.4740 .9970 1.4740 .9970

1.3 0.0

1 5.1200 19.4900

2 0.0 20.0 60.0 39.2 2 4

1 0.0 39.2 39.2 5 11

*Eddy, C. R., Showell, J. S., and Martin, V. G. Uses of digital computers in theoretical analytical chemistry. III. Some computational experiments on irregularities in countercurrent distribution. Journal of the American Oil Chemists' Society [in press].

- Card 1 Print the following transfers: 0-25, 50, 100,
150, 200, 250, 275, 300
- Card 2 Label the components CMX, CMY, CMZ
- Card 3 180 transfers, 10 tubes, 100 inputs
- Card 4 Densities: CMX, CMY, CMZ, most dense solvent,
least dense solvent
- Card 5 1.3 milliliters of upper phase retained on the wall
- Card 6 $NGOTO = 1$, 5.12 grams of solute per input,
19.49 milliliters of upper phase fed on each
operation
- Card 7-a Fill tubes 0-2 with 0.0 gram CMX,
20.0 grams CMZ, 60.0 grams CMY
Cut-off volume for tubes 0-2 is 39.2
- Card 7-b Fill tubes 3-9 with 39.2 milliliters of lower
phase pre-equilibrated with upper phase
Cut-off volume of tubes 3-9 is 39.2
- Card 8 When simulation is done, stop. Do not
proceed with another run.

Input for PUGET

The density card (card 4) is used as input for PUGET

Procedure for Preparing Subroutines

Experimental data for the upper and lower solubility curves must be reduced to (5th degree or lower order) polynomial relationships between X (the solute) and Y (the best solvent). In a Type I system, where there is only one solubility curve, an artificial division into upper and lower curves may be made at the plait point. The fit curves need only be accurate descriptions of the system within the concentration regions of interest in the program. In our acetone-water-chloroform system, (fig. 1), * only the solid parts of the solubility curve were fit. The area above tie line 1 was never reached in the simulation. Interpolation beyond this tie line would be difficult. Small errors near the plait point lead to large errors in relative phase volumes.

*Brancker, A. V., Hunter, T. G., and Nash, A. W. J. Phys. Chem. 44:
683-698. 1940.

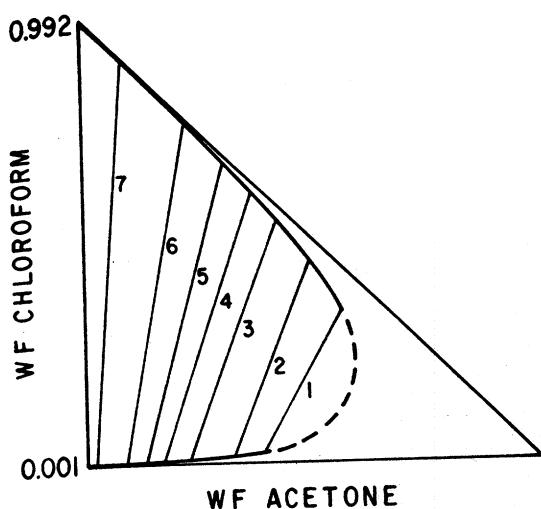


Figure 1.--Ternary diagram of acetone, water, chloroform system at 25° C.

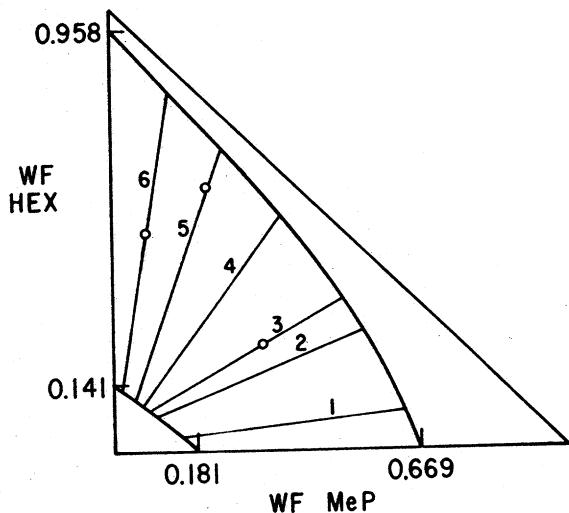


Figure 2.--Ternary diagram of methyl palmitate, hexane, acetonitrile system at 25° C.

The polynomials of the upper and lower curves are inserted in the appropriate subroutines. The coefficients of the upper curve are used in CURVU and KURVU. In CURVU the coefficients are expressed as an array: A(1) Y-intercept, A(2) coefficient of X, A(3) coefficient of X^2 , and so forth. In KURVU the coefficients are in standard form for evaluating a polynomial by Horner's rule*. The coefficients of the lower curve are similarly used in CURVL and KURVL. CURVU and CURVL in turn call NWRPH, which finds the roots of a polynomial by the Newton-Raphson method*. In CURVU the value of X_2 must be chosen according to the types of system. In a Type II (fig. 2) system X_2 is set equal to the X-intercept of the upper curve**. In a Type I system X_2 is the highest value of X on the upper curve which is of interest in the simulation. In figure 1 this corresponds to the point where tie line 1 intersects the upper curve.

The equation of each tie line is calculated. These equations are used in subroutine SBTIE. TIE(J, 1) is the intercept of tie line J, TIE(J, 2) is the slope of tie line J. The tie lines must be ordered in the array as shown on the ternary diagram, (fig. 1). The first two positions in the array TIE denote the intersections of the solubility curves with the x-axis. If there are no x-intercepts, the values 2.0, 2.0 are inserted in the first two positions of the TIE array. These values are tested by the mainline program to determine whether the system is Type I or Type II. The last two positions in the TIE array denote the intersections of the solubility curves with the y-axis.

*McCracken, D. D., and Dorn, W. S. Numerical methods and FORTRAN Programming. 1964. John Wiley & Sons, Inc. New York.

**Barford, R. A., Bertsch, R. J., and Rothbart, H. L. J. Am. Oil Chem. Soc. 45: 141-143. 1968.

In subroutine TUBEN, the volume of upper phase to be fed (after solute input is discontinued) must be set at the desired value. VUP is the name of the variable.

During execution of TERN2, the following switches can be utilized:

- '3' ends program without ability to restart
- '6' interrupts program with ability to restart
- '11' every transfer printed
- '12' all printout is suppressed

During execution of PUGET, switch '3' can be used to end printout when desired.

A note on the symbols used in the programs. These programs were originally written for a ternary system of methyl palmitate, hexane, acetonitrile. The letters MEP, HEX, ACN or simply M, H, A appear as part of the variable names throughout the programs. These letters in the general case refer to components X, Y, Z, respectively.

MAINLINE PROGRAM TERN1

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
*IOCS(CARD,1132PRINTER,DISK,TYPEWRITER)
** TERN1 - VERSION OF MAY 20, 1968
C   THIS PROGRAM SIMULATES A COUNTER CURRENT DISTRIBUTION, MAKING USE
C   OF THE TERNARY DIAGRAM THAT DESCRIBES THE THREE COMPONENT SYSTEM
C   IN WEIGHT FRACTIONS.
C   V. G. MARTIN, R. A. BARFORD
DIMENSION WTMEP(202),WTHEX(202),WTACN(202),TIE(20,2)
DIMENSION KPRNT(16), VCUT(202)
DIMENSION WDSK1(5), WDSK2(5), WDSK3(5)
COMMON DENSA,DENSH,DENSM,FTACN,FTHEX,FTMEP,GRAMP,DENLT,DENHV
COMMON I, IDENT, JR, JROUT, JRSGL, JSWCH, K, KDISK, KPRN
COMMON KPRNT, KPRON, KRITE
COMMON L, N, NDISK, NIGHT, NITE, NKIM, NLIM, NOFF, NOUT
COMMON NREC, NSTOR, NTIE, NTUBA, NTUBE, NTUBZ
COMMON RHOLO, RHOUP, TEM1, TEM2, TIE, TIEB
COMMON TIEM, TRYV, VCUT, VECUT, VIN, VLACN, VLHEX, VLMEP
COMMON VLOW, VLOW1, VUP, VUP1, WALLO, WALUP, WDSK1, WDSK2
COMMON WDSK3, WLMEP, WTACN, WTHEX, WTLOW, WTMEP, WTTRA
COMMON WTTRH, WTTRM, WTTOT, WTUPR, WUMEP, X, Y, YTACN, YTHEX
DEFINE FILE 1(9,320,U,KDISK)
DEFINE FILE 2(200,53,U,NREC)
1 FORMAT(A3,1X,A3,1X,A3)
5 FORMAT (' COUNTER CURRENT DISTRIBUTION BY PROGRAM TERN1, VERSION
510F MAY 20, 1968')
10 FORMAT (16I5)
15 FORMAT (5F 7.4,215)
20 FORMAT ('1 TUBE WEIGHT OF WEIGHT OF WEIGHT TOT WE
1IGHT TOT WEIGHT TOT VOLUME VOLUME TRANSFER'
2' NUMBER CMX IN',9X,'CMX IN',12X,'CMX',11X,'CMY',12X,'CMZ','11X
3,'UPPER',10X,'LOWER',6X,'NUMBER'/13X,'UPPER',10X,'LOWER')
35 FORMAT( 19X, 11, 4F10.4, 215 )
40 FORMAT(/28X,'-INITIAL CONDITIONS-')
41 FORMAT(30X,'UPPER',8X,'LOWER')
42 FORMAT(10X,'WEIGHT',A4,5X,F11.4,F13.4)
43 FORMAT(2F5.2)
44 FORMAT(10X,'WT FRCT',A4,4X,F11.4,F13.4)
45 FORMAT(10X,'VOLUME',9X,F11.4,F13.4)
      WRITE (3,5)
50 CALL SBTIE (TIE,NTIE)
C   GET THE SLOPES AND INTERCEPTS OF THE TIE LINES FROM A SUBROUTINE.
C----BLOCK 1 - READ VALUES FOR COMPUTATION FROM DISK OR CARDS.
100 READ (1'1) IDENT,DENSM,DENSH,DENSA,NIGHT,NREC,KPRON,KPRN,DENLT,
1DENHV,NSTOR,NLIM,NTUBE,NOFF,KRITE,NTUBA,WALUP,WALLO,KPRNT
      IF (IDENT - 8) 190,110,110
C   BEGINNING OF NEW COMPUTATION
110 READ (2,10) KPRNT
      WRITE(3,10) KPRNT
      IF(KPRNT(1)-8888) 115,210,115
115 NIGHT = 0
```

TERN1 (CONT.)

```
NSTOR = 1
KPRON = 2
KPRN = 1
KRITE = 2
IDENT = 1
C      EMPTY FILE 2 OF PREVIOUS CONTENTS.
DO 120 I = 1,5
WDSK1(I) = 0.0
WDSK2(I) = 0.0
120 WDSK3(I) = 0.0
DO 130 I = 1,200
130 WRITE (2'1) WDSK1, WDSK2, WDSK3, NIGHT
NREC = 1
C      READ BOUNDARY CONDITIONS FROM CARDS.
135 READ(2,1) CHEM1,CHEM2,CHEM3
READ (2,10) NLIM,NTUBE,NOFF
WRITE(3,10) NLIM,NTUBE,NOFF
138 READ (2,15) DENSM,DENSH,DENSA,DENHV,DENLT
C      DENSM IS THE DENSITY OF THE COMPONENT EXPRESSED AS THE INDEPENDENT
C      VARIABLE IN THE WEIGHT FRACTION SYSTEM.
C      DENSH IS THE DENSITY OF THE COMPONENT EXPRESSED AS THE DEPENDENT
C      VARIABLE IN THE WEIGHT FRACTION SYSTEM. IT IS THE BEST SOLVENT
C      FOR DENSM.
C      DENSA IS THE DENSITY OF THE COMPONENT EXPRESSED AS A FUNCTION OF
C      THE OTHER TWO COMPONENTS IN THE WEIGHT FRACTION SYSTEM.
C      WRITE(3,15) DENSM,DENSH,DENSA,DENHV,DENLT
C      DENLT AND DENHV ARE DENSITIES OF LEAST DENSE AND MOST DENSE
C      SOLVENTS RESPECTIVELY.
READ(2,43) WALUP,WALLO
WRITE(3,43) WALUP,WALLO
READ(2,35) NGOTO,XUP1,XUP2,XUP3,VIN
GO TO (140,145,150), NGOTO
140 VMEP = XUP1/DENSM
WTMEP(1) = XUP1
VIN = XUP2
VUP = VIN-VMEP
IF (DENSH-DENLT) 4040, 141, 142
141 RHOUP = 1./(TIE(NTIE,2)/DENSH+(1.-TIE(NTIE,2))/DENSA)
GO TO 143
142 TEMP=TIE(NTIE,2)
TIE(NTIE,2)=TIE(NTIE,1)
TIE(NTIE,1) = TEMP
GO TO 141
143 WTUPR = VUP*RHOUP
WTHEX(1) = TIE(NTIE,2) * WTUPR
WTACN(1) = WTUPR-WTHEX(1)
WTTOT = WTUPR + WTMEP(1)
144 FUMEP = WTMEP(1) / WTTOT
FUHEX = WTHEX(1) / WTTOT
FUACN = WTACN(1) / WTTOT
GO TO (155,146), NGOTO
```

TERN1 (CONT.)

```
145 WTMEP(1) = XUP1
      WTHEX(1) = XUP2
      WTACN(1) = XUP3
      WTTOT = WTMEP(1) + WTHEX(1) + WTACN(1)
      GO TO 144
146 RHOUP = 1. / (FUMEP/DENSM + FUHEX/DENSH + FUACN/DENSA )
      VIN = WTTOT / RHOUP
      GO TO 155
150 FUMEP = XUP1
      FUHEX = XUP2
      FUACN = XUP3
      RHOUP = 1. / (FUMEP/DENSM + FUHEX/DENSH + FUACN/DENSA )
      WTTOT = VIN * RHOUP
      WTMEP(1) = FUMEP * WTTOT
      WTHEX(1) = FUHEX * WTTOT
      WTACN(1) = FUACN * WTTOT
C     FILLING THE TWO HUNDRED TUBES AND SPECIFYING THEIR CUT OFF VOLUMES
C     IS ACCOMPLISHED BY STATEMENTS 140 TO 150.
155 READ(2,35) NGOTO,XL01,XL02,VLOW,VECUT,NTUBA, NTUBZ
      GO TO (156,157),NGOTO
156 GRAMP = XL01
      VLOW = VLOW - GRAMP/DENSM
      RHOLO = 1. / (TIE(NTIE,1) / DENSH + (1. - TIE(NTIE,1)) / DENSA )
      WTLOW = VLOW * RHOLO
      WTTOT = GRAMP + WTLOW
      YTHEX = TIE(NTIE,1) * WTLOW
      YTACN = WTTOT - YTHEX
      FLMEP = GRAMP / WTTOT
      FLHEX = YTHEX / WTTOT
      FLACN = YTACN / WTTOT
      GO TO 158
157 GRAMP = XL01
      YTHEX = VLOW
      YTACN = XL02
      WTTOT = GRAMP + YTHEX + YTACN
      FLACN = YTACN / WTTOT
      FLHEX = YTHEX / WTTOT
      FLMEP = GRAMP / WTTOT
158 DO 160 I = NTUBA,NTUBZ
      WTMEP(I) = GRAMP
      WTHEX(I) = YTHEX
      WTACN(I) = YTACN
160 VCUT(I) = VECUT
      IF(NTUBZ-NTUBE-1) 155, 161, 4040
161 IF(DENSH-DENLT) 4040, 170, 165
165 TEMP=TIE(NTIE,1)
      TIE(NTIE,1) = TIE(NTIE,2)
      TIE(NTIE,2)=TEMP
170 WRITE(3,40)
      WRITE(3,41)
      WRITE(3,42) CHEM1,WTMEP(1),GRAMP
```

TERN1 (CONT.)

```
      WRITE(3,42) CHEM2,WTHEX(1),YTHEX
      WRITE(3,42) CHEM3,WTACN(1),YTACN
      WRITE(3,44) CHEM1,FUMEP,FLMEP
      WRITE(3,44) CHEM2,FUHEX,FLHEX
      WRITE(3,44) CHEM3,FLACN,FLACN
      WRITE(3,45) VIN,VLOW
      GO TO 200
C     CONTINUE PREVIOUS COMPUTATION. GET VARIABLES FROM THE DISK.
190  READ (1'2) (WTMEP(I),I=1,101)
      READ (1'3) (WTMEP(I),I=102,202)
      READ (1'4) (WTHEX(I),I=1,101)
      READ (1'5) (WTHEX(I),I=102,202)
      READ (1'6) (WTACN(I),I=1,101)
      READ (1'7) (WTACN(I),I=102,202)
      READ (1'8) (VCUT(I),I = 1,101)
      READ (1'9) (VCUT(I), I = 102,202)
      READ (2'NREC)WDSK1,WDSK2,WDSK3,NDISK
      NREC = NREC-1
200  WRITE (3, 20)
      CALL LINK (TERN2)
210  CALL EXIT
4040 STOP 4040
      END
```

MAINLINE PROGRAM TERN2

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
*IOCS(1132PRINTER,DISK)
**TERN2 VERSION OF MAY 20, 1968
C     V. G. MARTIN, R. A. BARFORD
      REAL LINEA, LINEH
      DIMENSION WTMEP(202),WTHEX(202),WTACN(202),TIE(20,2)
      DIMENSION TEMPO(3)
      DIMENSION KPRNT(16), VCUT(202)
      DIMENSION WDSK1(5), WDSK2(5), WDSK3(5)
      COMMON DENSA,DENSH,DENSM,FTACN,FTHEX,FTMEP,GRAMP,DENLT,DENHV
      COMMON I, IDENT, JR, JROUT, JRSGL, JSWCH, K, KDISK, KPRN
      COMMON KPRNT, KPRON, KRIT
      COMMON L, N, NDISK, NIGHT, NITE, NKIM, NLIM, NOFF, NOUT
      COMMON NREC, NSTOR, NTIE, NTUBA, NTUBE, NTUBZ
      COMMON RHOLO, RHOUP, TEM1, TEM2, TIE, TIEB
      COMMON TIEM, TRY, VCUT, VECUT, VIN, VLACN, VLHEX, VLMEP
      COMMON VLOW, VLOW1, VUP, VUP1, WALLO, WALUP, WDSK1, WDSK2
      COMMON WDSK3, WLMEP, WTACN, WTHEX, WTLOW, WTMEP, WTTRA
      COMMON WTTRH, WTTRM, WTTOT, WTUPR, WUMEP, X, Y, YTACN, YTHEX
      DEFINE FILE 2(200,53,U,NREC)
20 FORMAT(1X,16,7E15.6,16)
C----BLOCK 2 - BEGIN TRANSFER NUMBER LOOP.          BEGIN N LOOP.
200 NKIM = NLIM+1
      DO 880 N=NSTOR,NKIM
```

TERN2 (CONT.)

C AUTOMATIC PRINT CONTROL.
GO TO (210,230),KPRON
210 IF(N-KPRNT(KPRN)-2) 300,300,220
220 KRITE = 2
KPRN = KPRN + 1
KPRON = 2
GO TO 300
230 IF (KPRN - 17) 240,300,4040
240 IF(N-KPRNT(KPRN)-2) 300,250,250
250 KRITE = 1
KPRN = KPRN + 1
KPRON = 1
C-----BLOCK 3 - BEGIN TUBE NUMBER LOOP.
300 JR = N+NTUBA-2
IF (JR - NTUBE - 1) 320,320,310
310 JR = NTUBE + 1
320 IF (JR-1) 4040, 650, 325
C BEGIN MAIN COMPUTATION.
325 IF (WTMEP(JR)-(.1E-08)) 326, 327, 327
326 WTMEP(JR) = 0.0
327 WTTOT = WTMEP(JR) + WTHEX(JR)+WTACN(JR)
FTMEP = WTMEP(JR) / WTTOT
FTHEX = WTHEX(JR) / WTTOT
FTACN = WTACN(JR) / WTTOT
TEMPO(1) = WTMEP(JR)
TEMPO(2) = WTHEX(JR)
TEMPO(3) = WTACN(JR)
C TEST TO SEE IF THE COMPOSITION IS ON ONE OF THE AXES, I.E. BINARY.
IF(FTMEP) 4040,330,360
330 IF(FTHEX-TIE(NTIE,1))520,520,340
340 IF(FTHEX-TIE(NTIE,2)) 350,520,520
C TIE(1,1),TIE(1,2),TIE(NTIE,1), AND TIE(NTIE,2) DO NOT CONTAIN THE
C SLOPE AND INTERCEPT AS IN THE REST OF THE TIE LINE MATRIX, BUT
C CONTAIN THE INTERSECTIONS OF THE SOLUBILITY CURVES WITH THE
C X AND Y AXES RESPECTIVELY. IF NO X-INTERCEPTS EXIST, ENTER
C 2.0 FOR TIE(1,1) AND TIE(1,2)
350 FUMEP = 0.0
FLMEP = 0.0
FUHEX = TIE(NTIE,2)
FLHEX = TIE(NTIE,1)
GO TO 500
370 IF(FTMEP-TIE(1,1)) 520,520,375
360 IF(FTHEX) 4040, 361, 385
361 IF(TIE(1,1)-1.0) 370, 370, 520
375 IF(FTMEP-TIE(1,2))380,520,520
380 FUHEX = 0.0
FLHEX = 0.0
FUMEP = TIE(1,2)
FLMEP = TIE(1,1)
GO TO 500
C TEST TO SEE IF THE COMPOSITION OF THE TUBE IS IN THE MISCELL
C REGION.

TERN2 (CONT.)

```
385 CALL KURVL (FTMEP,Y)
C ANY POINT BELOW Y, WHICH IS ON THE LOWER SOLUBILITY CURVE, IS IN
C THE MISCELLY REGION.
IF(Y-FTHEX) 390, 520,520
390 CALL KURVU(FTMEP,Y)
C ANY POINT ABOVE Y, WHICH IS ON THE UPPER SOLUBILITY CURVE, IS IN
C THE MISCELLY REGION.
IF(FTHEX-Y) 395,520,520
395 J = 1
C-----BLOCK 4 - CALCULATE THE SLOPE AND INTERCEPT OF THE TIE LINE FOR
C TUBE JR.
400 J=J+1
TRYY = TIE(J,1) + TIE(J,2)*FTMEP
TEMP = FTHEX - TRYY
IF(ABS(TEMP)-.001) 460,405,405
405 IF (FTHEX - TRYY) 410, 460,470
C BRANCH TO STATEMENT 410 WHEN THE FIRST TIE LINE IS REACHED BELOW
C WHICH THE COMPOSITION LIES.
410 IF (J-2) 4040, 415, 430
415 IF (TIE(1,1)-1.0) 420, 4050, 4050
C STATEMENT 4040 IS A CATCHALL FOR EVERY SEEMINGLY IMPOSSIBLE
C SITUATION.
420 X = -TIE(J,1)/TIE(J,2)
GO TO 440
C CALCULATE THE INTERSECTION OF TIE LINES J AND J-1.
430 L = J-1
TEMP=TIE(L,2)-TIE(J,2)
X = (TIE(J,1) - TIE(L,1))/TEMP
440 Y = TIE(J,1) + TIE(J,2)*X
C CALCULATE A NEW TIE LINE.
TEMP=FTMEP-X
TIEM=(FTHEX-Y)/TEMP
TEMP=TIEM*FTMEP
TIEB=FTHEX-TEMP
C FIND THE INTERSECTION OF THE NEW TIE LINE WITH THE UPPER AND LOWER
C SOLUBILITY CURVES. TO GET THE WEIGHT FRACTIONS IN THE UPPER AND
C LOWER PHASES.
450 CALL CURVU(TIEM,TIEB,FTMEP,FUMEP,FUHEX)
CALL CURVL(TIEM,TIEB,FTMEP,FLMEP,FLHEX)
C CURVL AND CURVU WILL IN TURN CALL SUBROUTINE NWRPH IF THE
C SOLUBILITY CURVES ARE EXPRESSED BY POLYNOMIALS HIGHER THAN THE
C SECOND DEGREE. NWRPH FINDS THE INTERSECTION OF TWO FUNCTIONS BY
C THE TRIAL AND ERROR METHOD. THIS IS NOT AN EXACT SOLUTION AND FOR
C THIS PROGRAM THE FIRST VALUE FOUND WITHIN .0001 OF THE ACTUAL
C INTERSECTION IS RETURNED AS THE CORRECT VALUE.
GO TO 500
460 TIEM = TIE(J,2)
TIEB = TIE(J,1)
GO TO 450
470 IF(J-NTIE+1) 400,480,4040
C BRANCH TO 480 OCCURS WHEN THE POINT LIES BETWEEN THE UPPER-MOST
C TIE LINE AND THE Y-AXIS.
```

TERN2 (CONT.)

```
480 X = 0.0
      GO TO 440
C-----BLOCK 5 - CALCULATE THE WEIGHT OF THE UPPER AND LOWER PHASE.
500 FUACN=1.-FUMEP-FUHEX
      FLACN=1.-FLMEP-FLHEX
C      CALCULATE DENSITIES OF UPPER AND LOWER PHASES.
      TEMP=FUMEP/DENSM+FUHEX/DENSH+FUACN/DENSA
      RHOUP=1./TEMP
      TEMP=FLMEP/DENSM+FLHEX/DENSH+FLACN/DENSA
      RHOLO=1./TEMP
      IF(RHOLO - RHOUP) 510,520,530
C      BRANCH TO 510 OCCURS IF CONCENTRATIONS ARE ABOVE THE ISOPYCNIC
C      REGION.
510 TEMP = FLMEP
      FLMEP = FUMEP
      FUMEP = TEMP
      TEMP = FLHEX
      FLHEX = FUHEX
      FUHEX = TEMP
      TEMP = FLACN
      FLACN = FUACN
      FUACN = TEMP
      TEMP = RHOUP
      RHOUP = RHOLO
      RHOLO = TEMP
      GO TO 530
C      BRANCH TO 520 OCCURS IF IN THE ISOPYCNIC REGION, OR IF IN A
C      MISCELLY REGION.
520 FLMEP = FTMEP
      FUMEP = FLMEP
      FLHEX = FTHEX
      FUHEX = FLHEX
      FLACN = FTACN
      FUACN = FLACN
      TEMP=FLMEP/DENSM+FLHEX/DENSH+FLACN/DENSA
      RHOLO=1./TEMP
      WTLOW = WTTOT
      GO TO 600
530 AC = FTMEP - FLMEP
      BC = FTHEX - FLHEX
      LINEA = SQRT(AC*AC + BC*BC)
      AC = FUMEP - FTMEP
      BC = FUHEX - FTHEX
      LINEH = SQRT(AC*AC + BC*BC)
      TEMP = LINEA + LINEH
      LINEH = LINEH / TEMP
      WTLOW = LINEH * WTTOT
C-----BLOCK 6 - CALCULATE THE WEIGHT OF EACH COMPONENT THAT IS TO BE
C      TRANSFERED TO TUBE JR+1, AND WHAT REMAINS IN TUBE JR.
600 WTUPR = WTTOT - WTLOW
      VLOW = WTLOW/RHOLO
      VUP = WTUPR/RHOUP
```

TERN2 (CONT.)

```
    IF(VUP-WALUP) 1601, 1601, 602
1601 TEMP = WALUP
      WALUP = VUP
      VUP1 = VUP - WALUP
      WALUP = TEMP
      GO TO 1604
  602 VUP1 = VUP - WALUP
1604 VLOW1 = VLOW - WALLO
      IF ( VLOW1 + VUP1 - VCUT(JR)) 603, 603, 605
  603 TEM1 = 0.0
      GO TO 635
  605 IF ( VCUT(JR) - VLOW1) 610, 620, 630
C THIS TESTS WHETHER THE INTERFACE IS ABOVE OR BELOW THE SIDE ARM.
  610 TEM1 = VUP1/VUP*WTUPR
      TEM2 = (VLOW1-VCUT(JR))/VLOW*WTLOW
      GO TO 640
  620 TEM1 = VUP1/VUP*WTUPR
      TEM2 = 0.0
      GO TO 640
  630 TEM1 = (VUP1-VCUT(JR)+VLOW1)/VUP*WTUPR
  635 TEM2 = 0.0
  640 WTTRM = FUMEP*TEM1 + FLMEP*TEM2
      WTTRH = FUHEX*TEM1 + FLHEX*TEM2
      WTTRA = FUACN*TEM1 + FLACN*TEM2
      JRSGL = 2
      WTMEP(JR) = WTMEP(JR) - WTTRM
      WTHEX(JR) = WTHEX(JR)-WTTRH
      WTACN(JR) = WTACN(JR)-WTTRA
      GO TO 660
  650 WTTRM = WTMEP(1)
      WTTRH = WTHEX(1)
      WTTRA = WTACN(1)
      JRSGL = 1
  660 WTMEP(JR+1) = WTMEP(JR+1) + WTTRM
      WTHEX(JR+1) = WTHEX(JR+1) + WTTRH
      WTACN(JR+1) = WTACN(JR+1) + WTTRA
C-----BLOCK 7 - PRINT CONTROL
  705 CALL DATSW(12,JSWCH)
      GO TO (745,710),JSWCH
  710 CALL DATSW (11,JSWCH)
      GO TO (730,720),JSWCH
  720 GO TO (730,745),KRITE
  730 GO TO (800,740),JRSGL
  740 WUMEP = FUMEP*WTUPR
      WLMEP = FLMEP*WTLOW
      NOUT = N-2
      JRROUT = JR-2
      WRITE(3,20) JRROUT,WUMEP,WLMEP,TEMPO,VUP,VLOW,NOUT
  745 IF(JR-NTUBE)800,800,750
  750 IF(WTTRM+WTTRH+WTTRA) 4040,800,760
  760 IF(NIGHT) 4040,770,780
  770 NDISK = N
```

TERN2 (CONT.)

```

780 NIGHT = NIGHT + 1
WDSK1(NIGHT)=WTTRM
WDSK2(NIGHT) = WTTRH
WDSK3 (NIGHT)=WTTRA
IF (NIGHT - 5) 800,790,4040
790 WRITE(2'NREC)WDSK1,WDSK2,WDSK3,NDISK
NIGHT = 0
C-----BLOCK 8 - RECYCLING OF THE JR-LOOP.
800 CALL DATSW(3,JSWCH)
GO TO (810,820),JSWCH
810 IDENT = 8
GO TO 900
820 IF(JR-1) 4040,840,830
830 JR = JR-1
GO TO 320
C      END OF JR LOOP.                                END OF JR-LOOP.
840 IF(N-NOFF) 860,850,860
850 CALL TUBEN
C      SUBROUTINE TUBEN CONTROLS SOLUTE INPUT TERMINATION.
860 CALL DATSW(6,JSWCH)
GO TO (870,880),JSWCH
870 NSTOR = N+1
GO TO 900
880 CONTINUE                                         END OF N-LOOP.
C      END OF N-LOOP.
IDENT = 9
900 CALL LINK (TERN3)
4040 STOP 4040
4050 WRITE(3,4051)
4051 FORMAT('OCONC. IS BEYOND LAST TIE LINE.')
4052 GO TO 4040
END

```

MAINLINE PROGRAM TERN3

```

*ONE WORD INTEGERS
*EXTENDED PRECISION
*IOCS(CARD,1132PRINTER,DISK)
** TERN3 - VERSION OF MAY 20, 1968
C      V. G. MARTIN, R. A. BARFORD
DIMENSION WTMEP(202),WTHEX(202),WTACN(202),TIE(20,2)
DIMENSION KPRNT(16), VCUT(202)
DIMENSION WDSK1(5), WDSK2(5), WDSK3(5)
COMMON DENSA,DENS, DENS, FTACN,FTHEX,FTMEP,GRAMP,DENLT,DENHV
COMMON I, IDENT, JR, JROUT, JRSGL, JSWCH, K, KDISK, KPRN
COMMON KPRNT, KPRON, KRITE
COMMON L, N, NDISK, NIGHT, NITE, NKIM, NLIM, NOFF, NOUT
COMMON NREC, NSTOR, NTIE, NTUBA, NTUBE, NTUBZ
COMMON RHOLO, RHOU, TEM1, TEM2, TIE, TIEB
COMMON TIEM, TRY, VCUT, VECUT, VIN, VLACN, VLHEX, VLMEP

```

TERN3 (CONT.)

```
COMMON VLOW, VLOW1, VUP, VUP1, WALLO, WALUP, WDSK1, WDSK2
COMMON WDSK3, WLMEP, WTACN, WTHEX, WTLOW, WTMEP, WTTRA
COMMON WTTRH, WTTRM, WTTOT, WTUPR, WUMEP, X, Y, YTACN, YTHEX
DEFINE FILE 1(9,320,U,KDISK)
DEFINE FILE 2(200,53,U,NREC)

25 FORMAT(//17I7)
C-----BLOCK 9 - SAVE DATA ON DISK.
900 IF(NIGHT)4040,930,910
910 NITE = NIGHT+1
DO 920 J = NITE,5
WDSK1(J) = 0.0
WDSK2(J)= 0.0
920 WDSK3(J) = 0.0
WRITE(2'NREC) WDSK1,WDSK2,WDSK3,NDISK
NREC = NREC - 1
930 WRITE(1'1) IDENT,DENSM,DENS,DENSA,NIGHT,NREC,KPRN,KPRN,DENLT,
1DENHV,NSTOR,NLIM,NTUBE,NOFF,KRITE,NTUBA ,WALUP,WALLO,KPRNT
WRITE(3,25)IDENT,KPRN,KPRN,NIGHT,NREC,KRITE,NSTOR,NLIM,NOFF,NTUBE
WRITE(1'2) (WTMEP(I),I=1,101)
WRITE(1'3) (WTHEX(I),I=102,202)
WRITE(1'4) (WTACN(I),I=1,101)
WRITE(1'5) (WTACN(I),I=102,202)
WRITE(1'6) (WTACN(I),I=1,101)
WRITE(1'7) (WTACN(I),I=102,202)
WRITE(1'8) (VCUT(I),I = 1,101)
WRITE(1'9) (VCUT(I), I = 102,202)
IF(IDENT-8) 940, 50,940
50 CALL LINK (TERN1)
940 CALL EXIT
4040 STOP 4040
END
```

SUBROUTINE TUBEN

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
** TUBEN - VERSION OF OCTOBER 27, 1967
C      V. G. MARTIN
SUBROUTINE TUBEN
DIMENSION WTMEP(202),WTHEX(202),WTACN(202),TIE(20,2)
DIMENSION KPRNT(16), VCUT(202)
DIMENSION WDSK1(5), WDSK2(5), WDSK3(5)
COMMON DENSA,DENS,DENSM,FTACN,FTHEX,FTMEP,GRAMP,DENLT,DENHV
COMMON I, IDENT, JR, JROUT, JRSGL, JSWCH, K, KDISK, KPRN
COMMON KPRNT, KPRN, KRITE
COMMON L, N, NDISK, NIGHT, NITE, NKIM, NLIM, NOFF, NOUT
COMMON NREC, NSTOR, NTIE, NTUBA, NTUBE, NTUBZ
COMMON RHOLO, RHOUP, TEM1, TEM2, TIE, TIEB
COMMON TIEM, TRY, VCUT, VECUT, VIN, VLACN, VLHEX, VLMEP
COMMON VLOW, VLOW1, VUP, VUP1, WALLO, WALUP, WDSK1, WDSK2
```

TUBEN (CONT.)

COMMON WDSK3, WLMEP, WTACN, WTHEX, WTLOW, WTMEP, WTTRA
COMMON WTTRH, WTTRM, WTTOT, WTUPR, WUMEP, X, Y, YTACN, YTHER
C VUP IS THE VOLUME OF PRE-EQUILIBRATED UPPER PHASE TO BE FED IN
C AFTER SOLUTE INPUT IS SHUT OFF.
C VUP=20.35
IF(DENSH-DENLT) 4040, 101, 102
101 TEMP=TIE(NTIE,2)/DENSH+(1.-TIE(NTIE,2))/Densa
GO TO 103
102 TEMP=TIE(NTIE,2)
TIE(NTIE,2)=TIE(NTIE,1)
TIE(NTIE,1)=TEMP
GO TO 101
103 RHOUP=1./TEMP
WTUPR = VUP*RHOUP
WTMEP(1) = 0.0
WTHEX(1) = TIE(NTIE,2)*WTUPR
WTACN(1) = WTUPR - WTHEX(1)
IF(DENSH-DENLT) 4040, 106, 105
105 TEMP=TIE(NTIE,2)
TIE(NTIE,2)=TIE(NTIE,1)
TIE(NTIE,1)=TEMP
106 CONTINUE
RETURN
4040 STOP 4040
END

SUBROUTINE NWRPH

*ONE WORD INTEGERS

*EXTENDED PRECISION

** NWRPH VERSION OF SEPTEMBER 21, 1967 V.G.MARTIN

SUBROUTINE NWRPH (X,COEFF,X1,X2)

DIMENSION COEFF(6)

A0 = COEFF(1)

A1 = COEFF(2)

A2 = COEFF(3)

A3 = COEFF(4)

A4 = COEFF(5)

A5 = COEFF(6)

1 X3 = (X1 + X2) / 2.

X = X3

Y = (((A5*X+A4)*X+A3)*X+A2)*X+A1)*X+A0

IF(ABS(Y) - .0001) 5,2,2

2 IF (Y) 3,5,4

3 X2 = X3

GO TO 1

4 X1 = X3

GO TO 1

5 RETURN

END

THE NEXT FIVE SUBROUTINES DESCRIBE THE TYPE I SYSTEM IN FIG.(1)

SUBROUTINE CURVU TYPE I

```
*EXTENDED PRECISION
*ONE WORD INTEGERS
C      R. A. BARFORD
      SUBROUTINE CURVU(XM,XB,FTX,X,Y)
      DIMENSION A(6)
      A(1)=0.99444885-XB
      A(2)=-1.1646217-XM
      A(3)=0.9169423
      A(4)=-1.4994492
      A(5)=0.0
      A(6)=0.0
      X1=FTX
      X2=0.570
      CALL NWRPH(X,A,X1,X2)
      Y=XM*X+XB
      RETURN
      END
```

SUBROUTINE CURVL TYPE I

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      R. A. BARFORD
      SUBROUTINE CURVL(XM,XB,FTX,X,Y)
      DIMENSION A(6)
      A(1)=0.005733914-XB
      A(2)=0.090264683-XM
      A(3)=-.51693205
      A(4)=1.0952125
      A(5)=0.0
      A(6)=0.0
      X1=0
      X2=FTX
      CALL NWRPH(X,A,X1,X2)
      Y=XM*X+XB
      RETURN
      END
```

SUBROUTINE KURVU TYPE I

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      R. A. BARFORD
      SUBROUTINE KURVU(X,Y)
C      UPPER CURVE FOR ACETONE,H2O,HCCL3
      Y=(-1.4994492*X+0.9169423)*X-1.1646217)*X+0.99444885
      RETURN
      END
```

SUBROUTINE KURVL TYPE I

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      R. A. BARFORD
      SUBROUTINE KURVL(X,Y)
C      LOWER CURVE FOR ACETONE,H2O,HCCL3
      Y=((1.0952125*X-0.51693205)*X+0.090264683)*X+0.005733914
      RETURN
      END
```

SUBROUTINE SBTIE TYPE I

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      R. A. BARFORD
      SUBROUTINE SBTIE(TIE,NTIE)
      DIMENSION TIE(20,2)
      TIE(1,1) = 2.0
      TIE(1,2) = 2.0
      TIE(2,1) = -1.041
      TIE(2,2) = 2.440
      TIE(3,1) = -0.7148
      TIE(3,2) = 2.306
      TIE(4,1) = -0.5583
      TIE(4,2) = 2.608
      TIE(5,1) = -0.4773
      TIE(5,2) = 2.835
      TIE(6,1) = -0.4586
      TIE(6,2) = 3.508
      TIE(7,1) = -0.3858
      TIE(7,2) = 4.792
      TIE(8,1) = -0.4350
      TIE(8,2) = 14.8330
      TIE(9,1) = .006
      TIE(9,2) = .992
      C      NTIE IS THE NUMBER OF TIE LINES INCLUDING THE BINARY AXIS POINTS.
      NTIE = 9
      RETURN
      END
```

THE NEXT FIVE SUBROUTINES DESCRIBE THE TYPE II SYSTEM IN FIG.(2)

SUBROUTINE CURVU TYPE II

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      V.G. MARTIN
SUBROUTINE CURVU(XM,XB,FTX,X,Y)
DIMENSION A(6)
A(1) = .95797547 - XB
A(2) = -1.2440514 - XM
A(3) = 1.1377825
A(4) = -5.0599806
A(5) = 9.3194747
A(6) = -7.3607867
X1 = FTX
X2 = .669
CALL NWRPH (X,A,X1,X2)
Y = XM*X+XB
RETURN
END
```

SUBROUTINE CURVL TYPE II

```
*EXTENDED PRECISION
*ONE WORD INTEGERS
C      V.G. MARTIN
SUBROUTINE CURVL(XM,XB,FTX,X,Y)
DIMENSION A(6)
A(1) = .14126553 - XB
A(2) = -.55808087 - XM
A(3) = -1.2103756
A(4) = 0.0
A(5) = 0.0
A(6) = 0.0
X1 = 0.0
X2 = FTX
CALL NWRPH(X,A,X1,X2)
Y = XM * X + XB
RETURN
END
```

SUBROUTINE KURVU TYPE II

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      V.G. MARTIN
      SUBROUTINE KURVU(X,Y)
      Y = ((((-7.3607867 * X+9.3194747)*X-5.0599806)*X+1.1377825)*X
            1-1.2440514)*X+.95797547
      RETURN
      END
```

SUBROUTINE KURVL TYPE II

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      V.G. MARTIN
      SUBROUTINE KURVL(X,Y)
      Y = (-1.2103756*X - .55808087) *X+.14126553
      RETURN
      END
```

SUBROUTINE SBTIE TYPE II

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
C      V.G. MARTIN
      SUBROUTINE SBTIE(TIE,NTIE)
      DIMENSION TIE(20,2)
      TIE(1,1) = .181
      TIE(1,2) = .669
      TIE(2,1) = .0181
      TIE(2,2) = .1104
      TIE(3,1) = .0518
      TIE(3,2) = .4223
      TIE(4,1) = .0594
      TIE(4,2) = 1.294
      TIE(5,1) = .0648
      TIE(5,2) = 2.675
      TIE(6,1) = .0578
      TIE(6,2) = 6.090
      TIE(7,1) = .0678
      TIE(7,2) = 38.10
      TIE(8,1) = .14126553
      TIE(8,2) = .958
      NTIE = 8
      RETURN
      END
```

MAINLINE PROGRAM PUGET

```
*ONE WORD INTEGERS
*EXTENDED PRECISION
*IOCS(CARD,TYPEWRITER,KEYBOARD,1132PRINTER,PAPER TAPE,DISK)
** PUGET - VERSION OF MAY 20, 1968
C      V. G. MARTIN, R. A. BARFORD
      REAL LINEA,LINEH
      DIMENSION WTMEP(5),WTHEX(5),WTACN(5),TIE(20,2)
      DEFINE FILE 2(200,53,U,NREC)
101 FORMAT(14,2(E13.5,F7.3),4E13.5,2F12.3)
102 FORMAT(3F7.4)
103 FORMAT(2X,'NO.',2X,'WT-CMX IN',2X,'F-CMX',4X,'WT-CMX IN',2X,'F-CMX
1',4X,'WT-CMY IN',4X,'WT-CMY IN',4X,'WT-UPPER',5X,'WT-LOWER',8X,'VO
2LUME',6X,'VOLUME'/8X,'UPPER',5X,'UPPER',5X,'LOWER',5X,'LOWER',5X,
3UPPER',7X,'LOWER',37X,'UPPER',7X,'LOWER')
104 FORMAT('OTOTAL LOWER VOLUME WASHED OUT =',F9.4)
105 FORMAT('OTOTAL CMX ELUTED =',F9.4)
      WRITE(3,103)
      READ(2,102) DENSM,DENSH,DENSA
      TWMEP = 0.0
      TVLOW = 0.0
      NREC = 1
      CALL SBTIE(TIE,NTIE)
1 READ(2,'NREC)WTMEP,WTHEX,WTACN,NDISK
      DO 2 JR = 1,5
      XX = WTMEP(JR)+WTHEX(JR)+WTACN(JR)
      AA = WTMEP(JR)/XX + .0005
      BB = WTHEX(JR)/XX + .0005
      CC = WTACN(JR)/XX + .0005
325 IF(WTMEP(JR)-(.1E-08)) 326, 327, 327
326 WTMEP(JR) = 0.0
327 WTTOT = WTMEP(JR) + WTHEX(JR)+WTACN(JR)
      FTMEP = WTMEP(JR) / WTTOT
      FTHEX = WTHEX(JR) / WTTOT
      FTACN = WTACN(JR) / WTTOT
      IF(FTMEP) 4040,330,360
330 IF(FTHEX-TIE(NTIE,1))520,520,340
340 IF(FTHEX-TIE(NTIE,2)) 350,520,520
350 FUMEP = 0.0
      FLMEP = 0.0
      FUHEX = TIE(NTIE,2)
      FLHEX = TIE(NTIE,1)
      GO TO 500
370 IF(FTMEP-TIE(1,1)) 520,520,375
360 IF(FTHEX) 4040, 361, 385
361 IF(TIE(1,1)-1.0) 370, 370, 520
375 IF(FTMEP-TIE(1,2))380,520,520
380 FUHEX = 0.0
      FLHEX = 0.0
```

PUGET (CONT.)

```
FUMEP = TIE(1,2)
FLMEP = TIE(1,1)
GO TO 500
385 CALL KURVL(FTMEP,Y)
C ANY POINT BELOW Y, WHICH IS ON THE LOWER SOLUBILITY CURVE, IS IN
C THE MISCELLY REGION.
C IF(Y-FTHEX) 390, 520, 520
390 CALL KURVU(FTMEP,Y)
IF(FTHEX-Y) 395, 520, 520
395 J = 1
400 J=J+1
TRYY = TIE(J,1) + TIE(J,2)*FTMEP
TEMP = FTHEX - TRYY
IF(ABS(TEMP)-.001) 460, 405, 405
405 IF (FTHEX - TRYY) 410, 460, 470
410 IF (J-2) 4040, 415, 430
415 IF (TIE(1,1)-1.0) 420, 4040, 4040
420 X = -TIE(J,1)/TIE(J,2)
GO TO 440
430 L = J-1
TEMP=TIE(L,2)-TIE(J,2)
X = (TIE(J,1) - TIE(L,1))/TEMP
440 Y = TIE(J,1) + TIE(J,2)*X
TEMP=FTMEP-X
TIEM=(FTHEX-Y)/TEMP
TEMP=TIEM*FTMEP
TIEB=FTHEX-TEMP
450 CALL CURVU(TIEM,TIEB,FTMEP,FUMEP,FUHEX)
CALL CURVL(TIEM,TIEB,FTMEP,FLMEP,FLHEX)
GO TO 500
460 TIEM = TIE(J,2)
TIEB = TIE(J,1)
GO TO 450
470 IF(J-NTIE+1) 400, 480, 4040
480 X = 0.0
GO TO 440
500 FUACN=1.-FUMEP-FUHEX
FLACN=1.-FLMEP-FLHEX
TEMP=FUMEP/DENSM+FUHEX/DENSH+FUACN/DENSA
RHOUP=1./TEMP
TEMP=FLMEP/DENSM+FLHEX/DENSH+FLACN/DENSA
RHOL0=1./TEMP
IF(RHOL0 - RHOUP) 510, 520, 530
510 TEMP = FLMEP
FLMEP = FUMEP
FUMEP = TEMP
TEMP = FLHEX
FLHEX = FUHEX
```

PUGET (CONT.)

```
FUHEX = TEMP
TEMP = FLACN
FLACN = FUACN
FUACN = TEMP
TEMP = RHOUP
RHOUP = RHOLO
RHOLO = TEMP
GO TO 530
520 FLMEP = FTMEP
FUMEP = FLMEP
FLHEX = FTHEX
FUHEX = FLHEX
FLACN = FTACN
FUACN = FLACN
TEMP=FLMEP/DENSM+FLHEX/DENSH+FLACN/DENSA
RHOLO=1./TEMP
WTLOW = 0.0
FLMEP=0.0
GO TO 600
530 AC = FTMEP - FLMEP
BC = FTHEX - FLHEX
LINEA = SQRT(AC*AC + BC*BC)
AC = FUMEP - FTMEP
BC = FUHEX - FTHEX
LINEH = SQRT(AC*AC + BC*BC)
TEMP = LINEA + LINEH
LINEH = LINEH / TEMP
WTLOW = LINEH * WTTOT
600 WTUPR = WTTOT - WTLOW
WUMEP=WTUPR*FUMEP
WLMEP=WTLOW*FLMEP
WUHEX=WTUPR*FUHEX
WLHEX=WTLOW*FTHEX
VLOW = WTLOW/RHOLO
VUP = WTUPR/RHOUP
NFRCT=NDISK-1
WRITE(3,101)NFRCT,WUMEP,FUMEP,WLMEP,FLMEP,WUHEX,WLHEX,WTUPR,WTLOW,
1VUP,VLOW
TVLOW = TVLOW + VLOW
TWMEP = WTMEP(JR) + TWMEP
2 NDISK = NDISK+1
CALL DATSW (3,JSWCH)
GO TO (3,1),JSWCH
3 WRITE(3,104) TVLOW
WRITE(3,105) TWMEP
CALL EXIT
4040 STOP 4040
END
```